Dhirubhai Ambani
Institute of Information and Communication Technology

Employee Management System

Software Requirement Specifications (SRS)

nicher SaaS

2023

Prepared by:

Saurav Kapadiya
(202001033)

Abstract

This report includes a development presentation of an information system for managing the employee data within a small company or organization. The system as such as it has been developed is called Employee Management System.

Employers could gain insight into their employees via an employee management system, which allows them to better plan and manage work hours, lowering labor expenses and increasing productivity.

Nicher SaaS (Employee Management System) tries to solve every kind of problem that arises in a company, either that be employee related or HR related.

Table of Contents

1. Introduction

An employee management system is a software platform that helps employees' information, tasks,managing leaves and performance.

The primary goal of an employee management system is to streamline and automate employee related processes, reduce administrative workload, and enhance employee experience and productivity. It provides HR with real time access to relevant employee data, enabling them to make informed decisions and take timely actions.

The system also benefits employees by providing them with self-service tools to manage their personal information, schedule and benefits. It promotes transparency and communication between employees and HR departments, leading to better engagement and retention.

Overall, an employee management system is a critical tool for modern organizations that want to manage their human resources effectively, efficiently and compliantly. It helps them attract, retain and develop top talent while ensuring regulatory compliance and reducing HR-related risks.

2. Scope

Nicher SaaS (Employee Management System) will be used by HR managers, team leaders, and employees within the organization. It will provide a user-friendly interface with different access levels for different roles. The system will be developed using modern web technologies and will be accessible from any web-enabled device with appropriate authentication.

Better Employee Management is the main focus of the system. Scope of the project described in a modularized manner.

3. Objectives

- This project aims to simplify the task of maintaining records of the employees of the Company.
- To develop a well - designed database to store employee information.
- The objective of this project is to provide a comprehensive approach towards the management of employee information.
- System provides an efficient way to manage new employee details
- Acting as a coordinator to sort out the leave management issues

4. Description of working of the current system

The current system has done their day to day employee management work such as filing new employee details , categorizing the employee filed, applying for a leave, checking the leave balance and annual salary increment according to the performance.

An employee will fill a leave note which can be taken from the HR manager and the employee will hand that form to the HR manager after completion. Then the HR manager will write the leave information in the left log. Then at the end of the day the HR manager will check if that leave can be approved or not. If the leave can be approved, the HR manager will mark the leave as approved and put a sign.

Details of the employees are in separate files with the employee name.

5. Drawbacks of the existing system

Data entry, validation, and processing are done manually. This can be erroneous. Historical data cannot be systematically viewed or structured at any time. Not more than one person can access the data at the same time. Difficulties are arisen due to lack of analyzed data, finding the history data and lack of summary reports. Since the considerable amount of data was gathered, the difficulties of finding urgent and accurate information in a timely manner to make administrative decisions.

6. Modules and Design Goals

Modules

Employee information management, monitor working days and holidays , Set leave processes , salary status.

Design Goals

➔ Efficient Data Management:

 The system should be designed to efficiently manage employee data, including personal details, job roles, attendance history.

➔ Streamlined Processes:

 The system should automate routine processes such as onboarding, leave requests, and payroll, to reduce the burden on HR personnel and improve overall efficiency.

➔ Security and Privacy:

The system should have robust security features to protect employee data and prevent unauthorized access, as well as comply with data privacy regulations.

➔ User-Friendly Interface:

The system should have a user-friendly interface that is easy to navigate, intuitive and accessible to all users.

➔ Scaling:

Large number of employees might be using the system.

- Reliability
- Backward compatibility.
- Rapid Development.
- Understandability.
- Flexibility of requirements.

7. Requirements

7.1 Functional Requirements

- Login for different categories / level of users :

This feature is used by the user to login into the system. A user must login with his username and password to the system after registration.If they are invalid, the user is not allowed to enter the system.

➔ Username and password will be provided after user registration is confirmed.
➔ Passwords can be seen by the user after clicking on the eye button.

● Employee profile :

Employees must be able to view the system and their profile.

● Attendance management system :

The system offers scheduling and attendance tracking to ensure compliance with staffing needs.

● Leave application :

Employees can submit their requests and managers can review and approve them.

● Payroll management :

Payroll management helps administer accurate financial records of employees. It keeps track of their details- Salaries,team expenses.

● Expenses and allowances :

It helps organizations manage and track their employees' business related expenses and allowances and managers to review and approve them.

● Company information & profile :

A directory of all employees in the company, including their contact information, job title , department and location. Individual profile of each employee, containing their personal information.

7.2 Non-Functional Requirements

● Usability Requirement:

A The web application is designed for a user friendly environment and ease of use.

● User friendliness:

The proposed system should be easy to learn and use. It contains effective error handling, pleasant, easy to navigate through the GUI and many other UI guides for the user.

● Expandability:

The system should be designed to support future enhancement.

● Database Requirements:

NoSQL database that stores and retrieves the data at the highest level possible speed. Various models for data storage are required.

● Security Requirements :
  ❖ Password encryption is required for safety.
  ❖ Cookies or session generation for creating user specific features.
  ❖ Security and login for users.

● Implementation Requirements :

Implementation of the system using Flutter in front end, Node.js for backend and Passport.js for authentication and the database part is developed by MongoDB as NoSQL database.

- Database Security :

  Unauthorized person cannot access the panel and database and do not read and write the information.

- Availability :

  The website will be available for maximum possible time.

- Intuitive user experience and design :

  The user interface should be simple and intuitive, with easy to use navigation and clear labeling. Consistency in design key to creating an intuitive experience.

- Maximum Scalability :

  The system should be handled for a large number of employees with varying levels of access and permissions for different user roles.

- Robustness:

  The system should be able to cope with errors during the execution and needs to handle user input errors.

7.3 User Requirements :

User Categories:

Admin :

Has the highest level of access and control over the system.

Employee:

The lower layer of users, who get a few selected feature access and for only himself/herself.

Manager:

Higher level of users, who get a few selected features for other users too.

User Needs:

Employee

- ❖ View his/her profile.
- ❖ View his/her attendance history.
- ❖ View his/her performance metrics.
- ❖ View his/her salary status.
- ❖ Apply for leave and see the status of his/her leave application.
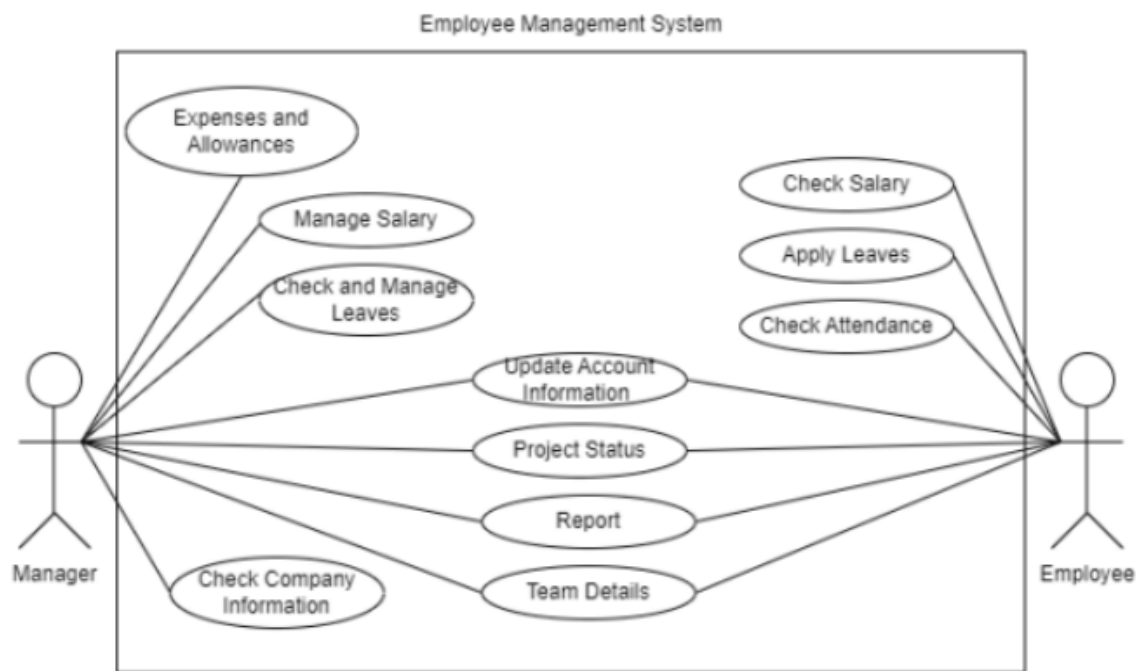- ❖ View department details

Manager

- ❖ Create an Employee profile.

- ❖ Approve or reject leave applications.

- ❖ View attendance of employees.

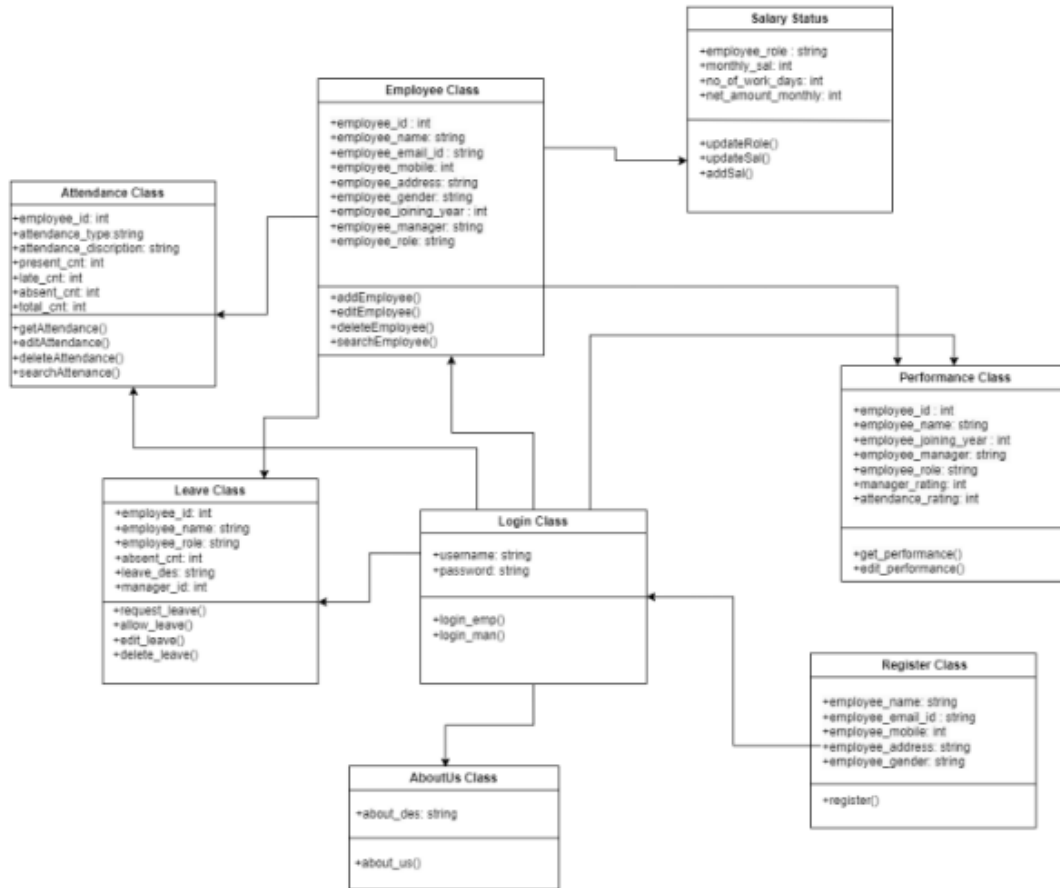- ❖ All other features and perks offered to employees

Use Case Model:

A use case model defines the various actions or tasks that users can perform within the system. For an employee management system, some potential use cases might include:

- Registering a new employee profile.
- Show employee profile page.
- Show salary status page.
- Managing the employee attendance history.
- Applying for a leave(leave application form).
- employee list (manager can see this)
- departments list (admin)
- logout

Use Case diagram :

Employee Management System

Class diagram :

Salary Status

+employee_role : string
+monthly_sal: int
+no_of_work_days: int
+net_amount_monthly: int

+updateRole()
+updateSal()
+addSal()

Employee Class

+employee_id : int
+employee_name: string
+employee_email_id : string
+employee_mobile: int
+employee_address: string
+employee_gender: string
+employee_joining_year : int
+employee_manager: string
+employee_role: string

+addEmployee()
+editEmployee()
+deleteEmployee()
+searchEmployee()

Attendance Class

+employee_id: int
+attendance_type:string
+attendance_discription: string
+present_cnt: int
+late_cnt: int
+absent_cnt: int
+total_cnt: int

+getAttendance()
+editAttendance()
+deleteAttendance()
+searchAttenance()

Performance Class

+employee_id : int
+employee_name: string
+employee_joining_year : int
+employee_manager: string
+employee_role: string
+manager_rating: int
+attendance_rating: int

+get_performance()
+edit_performance()

Leave Class

+employee_id: int
+employee_name: string
+employee_role: string
+absent_cnt: int
+leave_des: string
+manager_id: int

+request_leave()
+allow_leave()
+edit_leave()
+delete_leave()

Login Class

+username: string
+password: string

+login_emp()
+login_man()

Register Class

+employee_name: string
+employee_email_id : string
+employee_mobile: int
+employee_address: string
+employee_gender: string

+register()

AboutUs Class
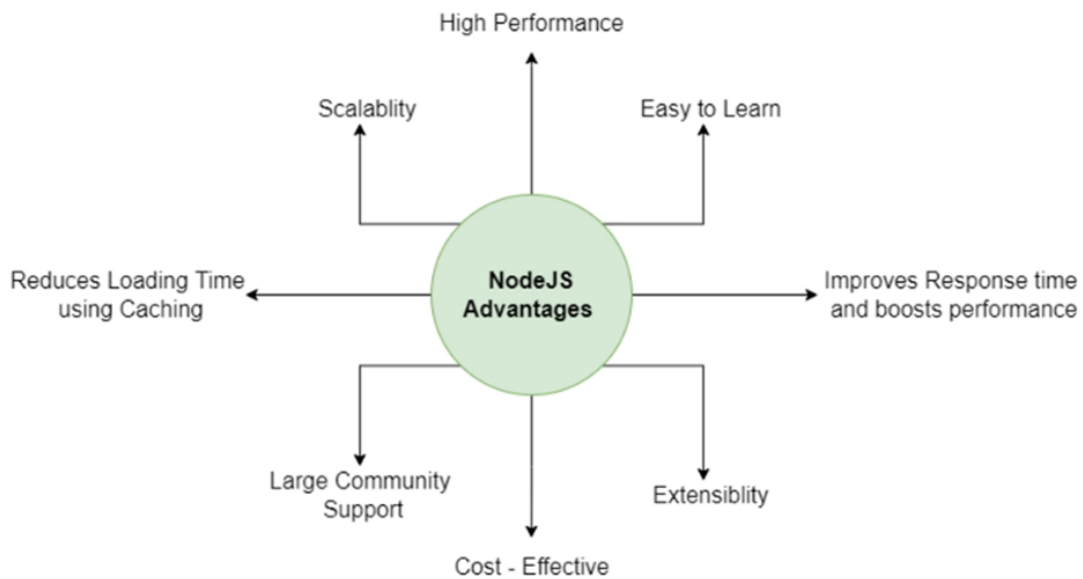
+about_des: string

+about_us()

8. Tools

- **Flutter for Frontend:**

  For the purposes of our project, we opt to work in frontend using Flutter language. Flutter is one of the best solutions to develop apps for Android and iOS, without having to write in a different codebase for each platform. The smartphone versions of these apps function as true, native apps on Apple and Android devices and are compiled for the respective platform before publication. They do not need a runtime module or a browser. Some of the benefits of flutter are:

  1. One codebase for all platforms

  2. "It's all Widgets" principle offers countless possibilities

  3. Rich libraries

  4. Fast testing with hot reload

- **Node.js for backend:**

  For the purpose of our project, we opt to work in the backend using Node.js language. Node.js is a JavaScript run-time environment built on Chrome's V8 engine, not a framework or library. It enables server-side execution of JavaScript code. Node.js is also an open-source platform, allowing greater freedom to develop real-time network applications. It provides asynchronous, event-driven I/O APIs to developers. All executions become non-blocking since it operates on a single-threaded event-based loop.
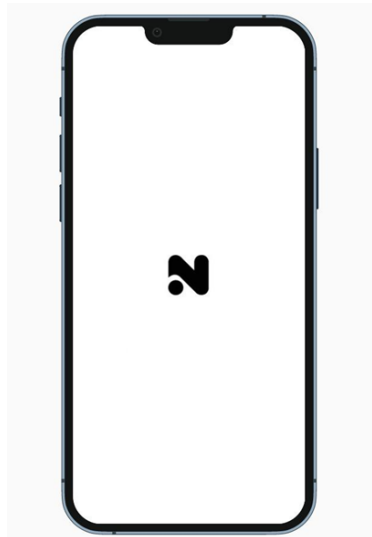
- **MongoDB as NoSQL database:**

  For the purpose of our project, we opt for MongoDB as a NoSQL database. Database selection plays a significant role in overall product development. How seamlessly you can edit, update, retrieve or delete depends on the database you choose. Of the two database types – non-relational and relational databases, you must choose the best fit based on your individual needs. You would have probably heard about the most in-demand database MongoDB, which is NoSQL and a very popular document database. In this article, we attempt to touch upon the reasons for the increased popularity of MongoDB.

  MongoDB, a document-based NoSQL database, is a schema-less database with compelling characteristics and salient features that allows users to query data in the most straightforward and tech-savvy way. The database supported with JSON-style storage enables users to manipulate and access data with no hassles.
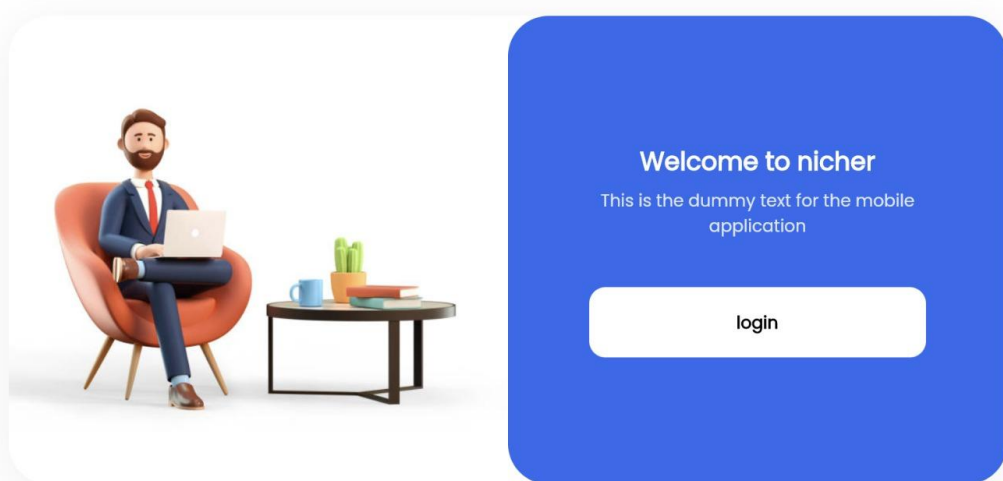
10. Implementation

- **Logo Page**

  After opening the application, the logo of Nicher SaaS will appear.
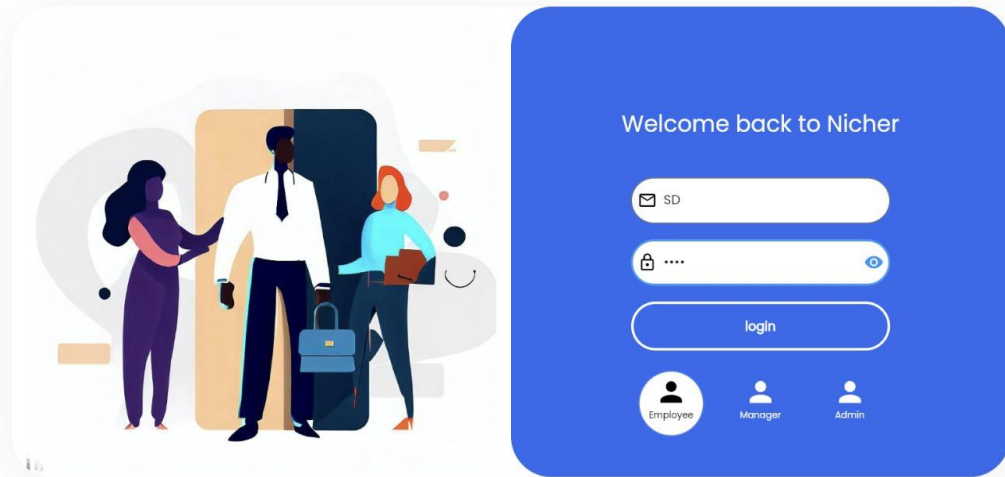

- **Welcome Page**

  After the logo page, the application displays a welcome page.

● **Login Page**
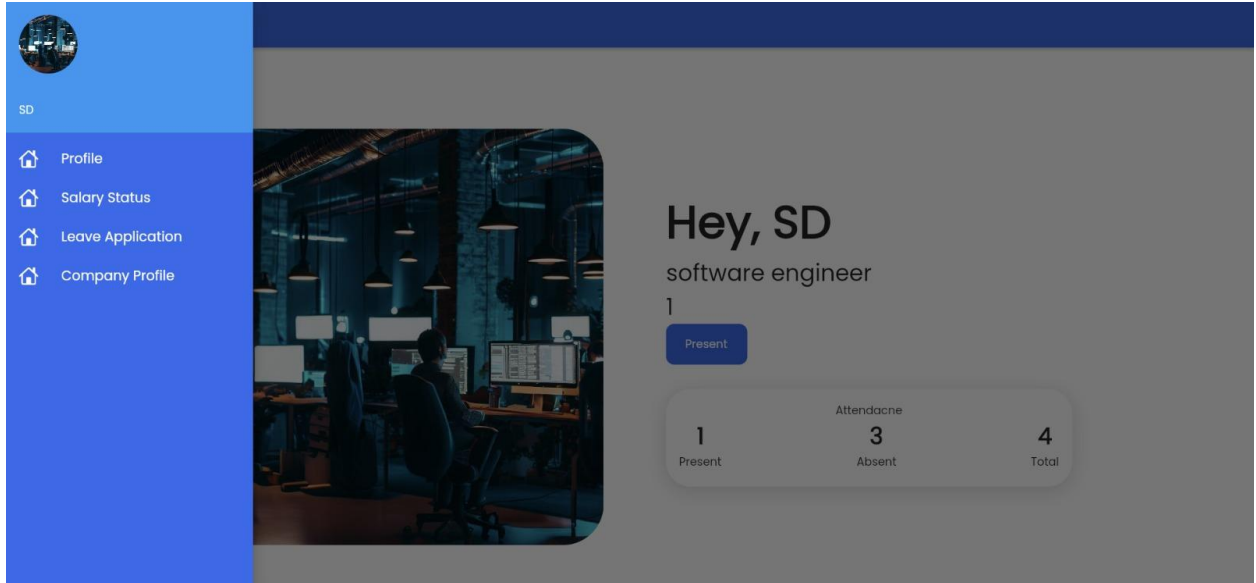
This is the very first user interface which came across to the user when this system is used and the login screen.The user has to give a username and a password that is already given to them .Privileges are varied from user to user.
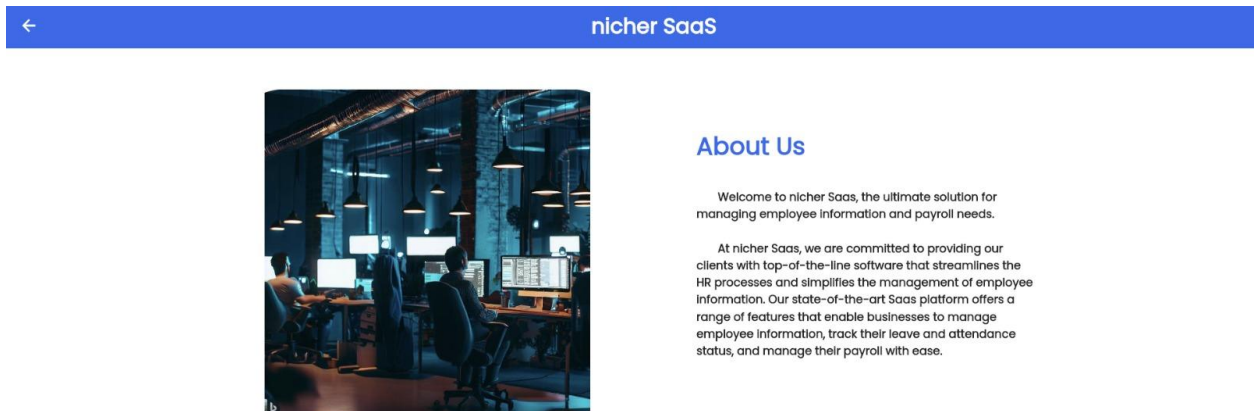

● **Homepage for employees**

This is the homepage shown when the user logged in identifying as an employee.

- **Company Profile Page**

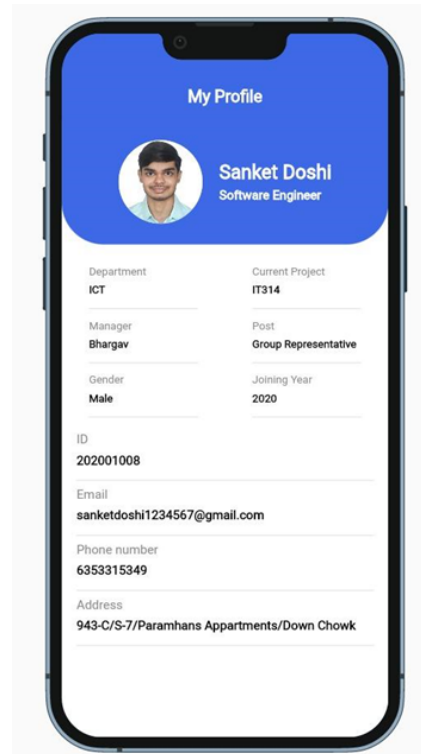 This is the company's profile page (About Us page).

- **Salary Status**

  If an employee wants to check the status of his/her salary, he/she can go or navigate to the salary page which is shown below.

Salary Status

Sanket Doshi
202001008

Post
Group Representative

Monthly Salary
1,25,000 INR

Number of absent days (in a month)
3

Amount to be credited
1,23,300 INR

- ● **Profile Page**

An employee or a manager can update or view his/her profile by navigating to the profile in the app(shown below)

● **Leave management**

2023-01-01T00:00:00.000Z To 2023-12-12T00:00:00.000Z
8 Medical

Appr ove   Reje ct

2023-04-20T00:00:00.000Z To 2023-04-23T00:00:00.000Z WFH	Approved
2023-04-20T00:00:00.000Z To 2023-04-23T00:00:00.000Z WFH	Approved
2020-12-12T00:00:00.000Z To 2020-12-12T00:00:00.000Z Medical	In Process
2020-12-12T00:00:00.000Z To 2020-12-12T00:00:00.000Z Medical	In Process
2022-01-01T00:00:00.000Z To 2022-02-01T00:00:00.000Z Medical	In Process
2022-01-01T00:00:00.000Z To 2022-02-01T00:00:00.000Z Medical	In Process
2023-04-26T00:00:00.000Z To 2023-04-27T00:00:00.000Z Medical	In Process
2023-04-26T00:00:00.000Z To 2023-04-26T00:00:00.000Z WFH	In Process
2023-04-28T00:00:00.000Z To 2023-04-28T00:00:00.000Z Parental	In P